TEXAS
The University of Texas at Austin

WHAT STARTS HERE CHANGES THE WORLD

# Prompt Engineering Tutorial

Jiawei Xu, Huimin Xu
jiaweixu@utexas.edu, huiminxu@utexas.edu
October 22, 2024

Google Drive Link:
https://drive.google.com/drive/folders/10h831vsebrPFCSf9NlFqJWR_4_W6g1Q4?usp=sharing

# Resources

- Slides and Notebook for this Tutorial
  - Adapted from ChatGPT Prompt Engineering for Developers.
    - Isa Fulford (OpenAI), Andrew Ng (deeplearning.ai)
  - OpenAI official prompt engineering guides.
- Other resources on prompt engineering / LLM reasoning.
  - LLM Reasoning: Key Ideas and Limitations. Denny Zhou (Google DeepMind)
  - Prompt Engineering Guide.
    - Chain-of-Thought
    - ReAct
    - ...

# [1hr Talk] Intro to Large Language Models – Andrej Karpathy

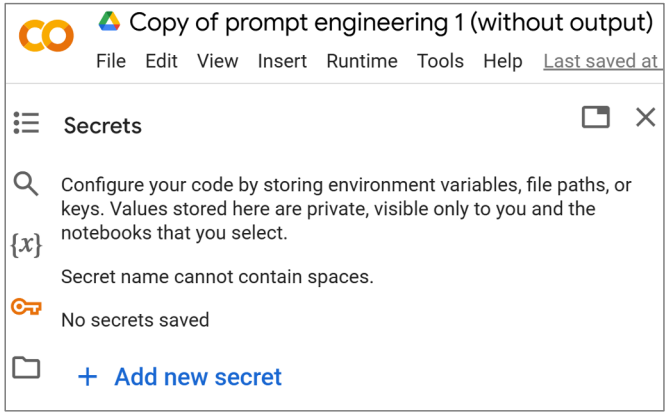
# Interact with Large Language Models (LLMs)

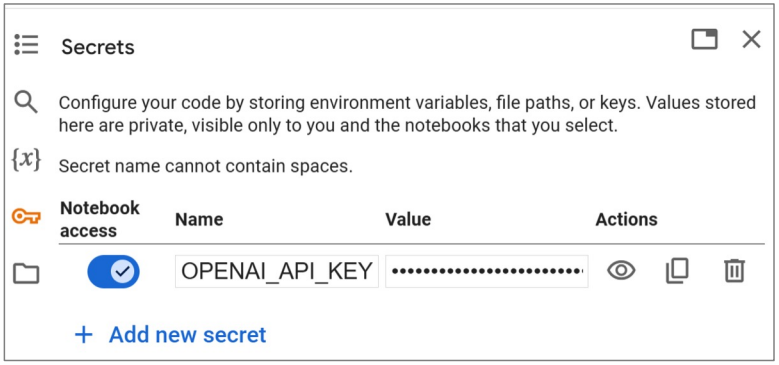
- Graphical User Interface (GUI)
  - chatgpt.com , meta.ai , claude.ai ...
  - for everyday use
- Application Programming Interface (API)
  - platform.openai.com, docs.anthropic.com , docs.mistral.ai/api/
  - for Programming
    - developers, data analysts, researchers ...
  - We will use the API extensively in this tutorial!

# Setup

- Google Colab
  - Link. Open the link and make a copy in your Google Drive so you can save it.
- Register your OpenAI key: link
  - Make sure you have $1 in your account.
  - Let me know if you have trouble with the API key.
- Add a secret in your Colab
  - Name: OPENAI_API_KEY
  - Value: 'sk-your_open_ai_key...'
- Let's play with the OpenAI API!
- Let me know if you have trouble with the API key.
  - OpenAI Key for this class (if you don't have one)

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



# Helper Function

```
# helper function
client = openai.OpenAI(api_key=openai.api_key)

def get_completion(prompt, model="gpt-3.5-turbo-1106", temperature=0):
    messages = [{"role": "user", "content": prompt}]
    response = client.chat.completions.create(
        model=model,
        messages=messages,
        temperature=temperature # this is the degree of randomness of the model's
output
    )
    return response.choices[0].message.content
```

```
prompt = 'hello world'
response = get_completion(prompt, temperature= 0)
```

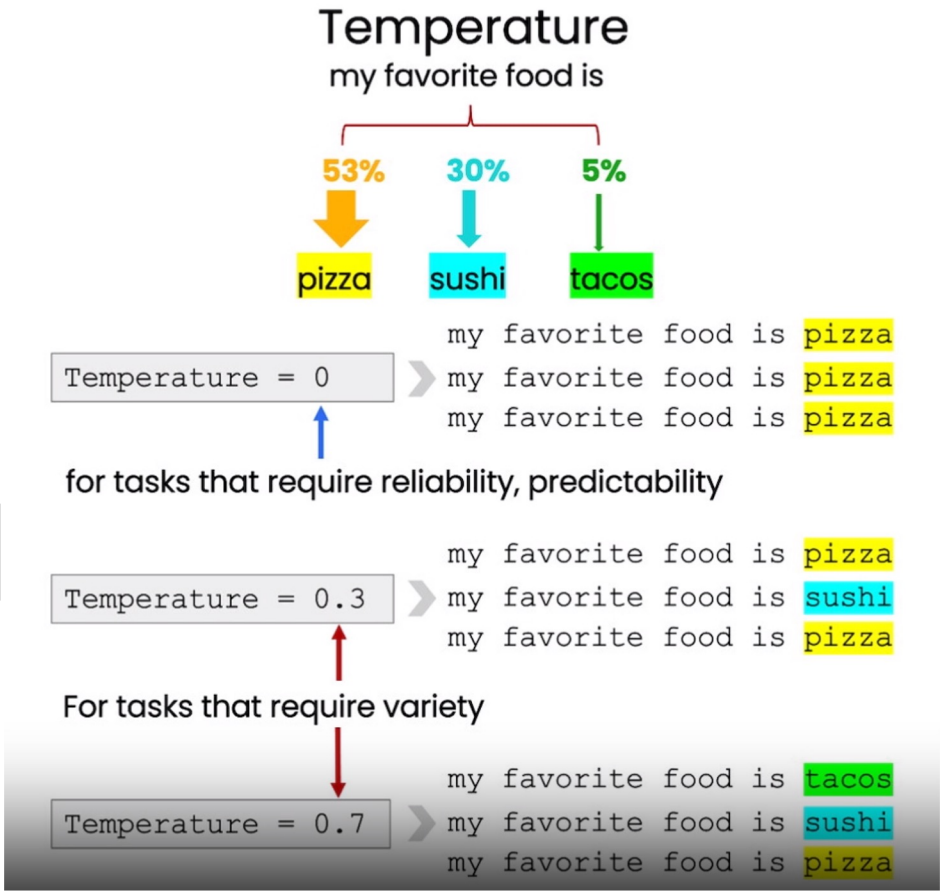
**Output from the API:**

Hello there! How can I assist you today?

# Temperature

**Temperature:** Controls randomness: Lowering results in less random completions. As the temperature approaches zero, the model will become deterministic and repetitive

```
prompt = 'My favorite food is'
response = get_completion(prompt, temperature=0)
```




Source:
ChatGPT Prompt Engineering for Developers

# 1.Prompting Principles

- Principle 1: Write clear and specific instructions
- Principle 2: Give the model time to “think”

# 1.Prompting Principles

- Principle 1: Write clear and specific instructions
  - Tactic 1: Use delimiters to clearly indicate distinct parts of the input
    - Delimiters can be anything like: ```, """, < >, <tag> </tag>
  - Tactic 2: Ask for a structured output
    - JSON and other format
  - Tactic 3: "Few-shot" prompting

# 1.Prompting Principles

- Principle 1: Write clear and specific instructions
- Principle 2: Give the model time to “think”
  - Specify the steps required to complete a task
  - Instruct the model to work out its own solution before rushing to a conclusion

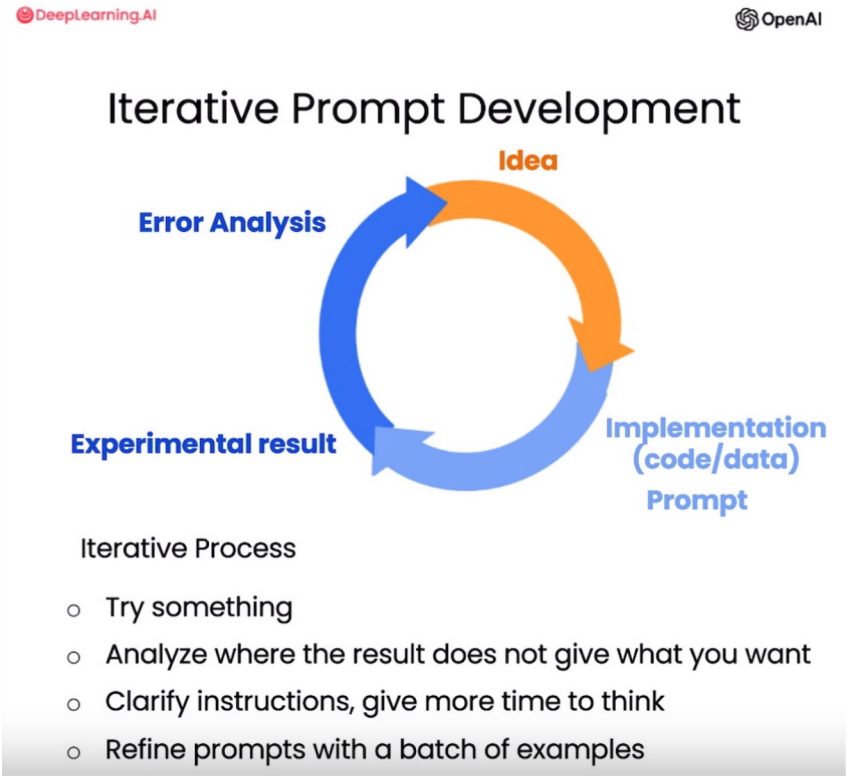
# 1.Prompting Principles

- Principle 1: Write clear and specific instructions
- Principle 2: Give the model time to “think”
- Model Limitations:
  - Hallucinations

# 2.Iterative Prompt Development

- Iteratively analyze and refine your prompts to generate marketing copy from a product fact sheet.

Source:
ChatGPT Prompt Engineering for Developers



# 3.Text Summarizing

- Summarize multiple product reviews

# 4.Inferring

- Sentiment (positive/negative)
- Identify types of emotions
- Inferring topics
- Make a news alert for certain topics

# 5.Transforming

- Translation
- Format Conversion

# 6. Let's build an OrderBot

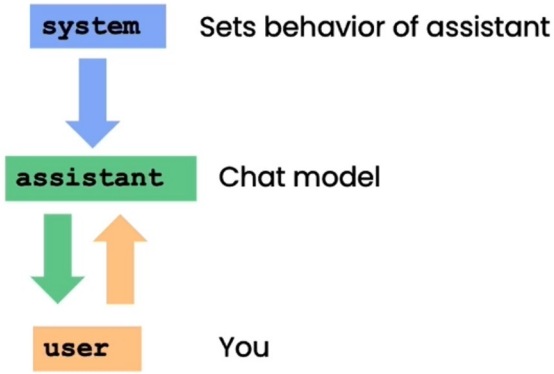
- We can automate the collection of user prompts and assistant responses to build an OrderBot. The OrderBot will take orders at a pizza restaurant.

# Multi-round conversations! The Chat Format

- context
- role

## System, User and Assistant Messages

```
messages =
[
 {"role": ”system",
  "content": "You are an assistant… "},
 {"role": "user",
  "content": "Tell me a joke "},
 {"role": ”assistant",
  "content": ”Why did the chicken… "},
 …
]
```



# Tokens

- Take the letters in lollipop and reverse them
  - ChatGPT: pilpolol
- Why?

```
[ ]  response = get_completion("Take the letters in lollipop \
     and reverse them")
     print(response)
```

```
pilpolol
```

"lollipop" in reverse should be "popillol"

```
[ ]  response = get_completion("""Take the letters in \
     l-o-l-l-i-p-o-p and reverse them""")
     response
```

```
'p-o-p-i-l-l-o-l'
```

Source: Building Systems with the ChatGPT API

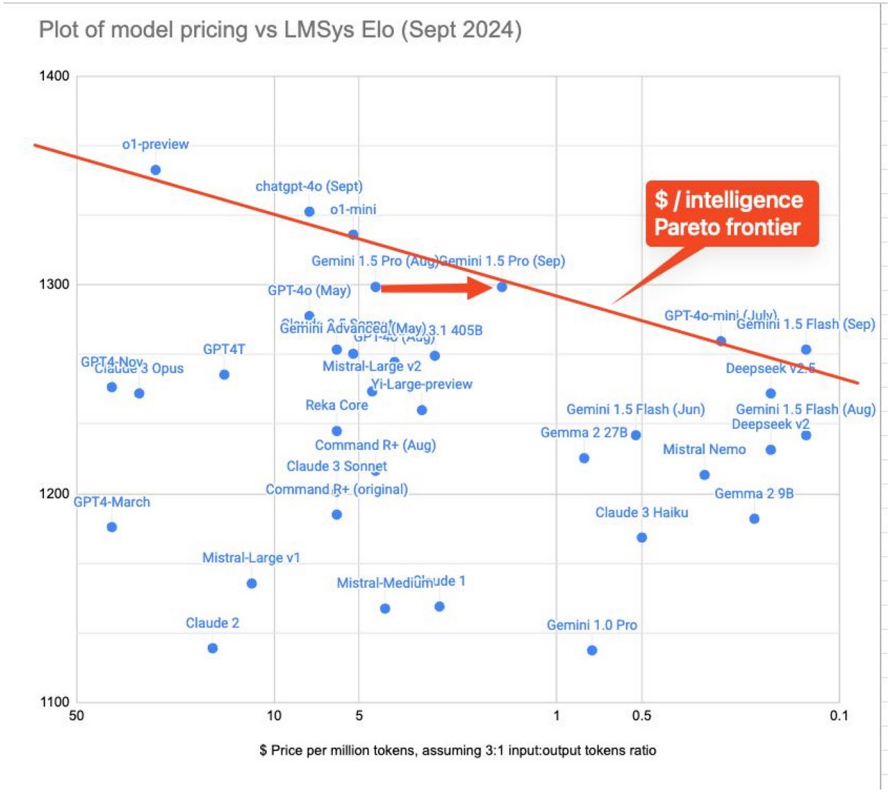
# Tokens

- Take the letters in lollipop and reverse them
  - ChatGPT: pilpolol
- Why?
  - The training unit in an LLM is a 'token,' not a letter.

Check the latest token limits:
https://platform.openai.com/docs/models

# LLM API Pricing

https://openai.com/api/pricing/



# LLM Arena

https://lmarena.ai/

| Rank* (UB) | Delta | Model | Arena Score | 95% CI | Votes | Organization | License |
|---|---|---|---|---|---|---|---|
| 1 | 0 | o1-preview | 1300 | +6/-4 | 17562 | OpenAI | Proprietary |
| 1 | 0 | ChatGPT-4o-latest (2024-09-03) | 1297 | +4/-4 | 28488 | OpenAI | Proprietary |
| 3 | 0 | Gemini-1.5-Pro-002 | 1271 | +5/-7 | 11430 | Google | Proprietary |
| 3 | 1 | Gemini-1.5-Pro-Exp-0827 | 1267 | +4/-3 | 32437 | Google | Proprietary |
| 3 | 0 | o1-mini | 1260 | +6/-5 | 17919 | OpenAI | Proprietary |
| 5 | 2 | GPT-4o-2024-05-13 | 1262 | +2/-3 | 99251 | OpenAI | Proprietary |
| 5 | 4 | Claude 3.5 Sonnet | 1259 | +2/-4 | 75957 | Anthropic | Proprietary |
| 7 | 3 | Gemini Advanced App (2024-05-14) | 1252 | +3/-3 | 52235 | Google | Proprietary |
| 7 | 2 | Meta-Llama-3.1-405b-Instruct-bf16 | 1251 | +5/-5 | 14496 | Meta | Llama 3.1 Community |
| 8 | 2 | Meta-Llama-3.1-405b-Instruct-fp8 | 1252 | +3/-4 | 39428 | Meta | Llama 3.1 Community |
| 8 | -2 | Grok-2-08-13 | 1249 | +4/-4 | 35661 | xAI | Proprietary |

# Have Fun with LLMs!